* VZ-LINK *
NEWLETTER

<< AUCKLAND VZ 300/200 USERS CLUB >>

ISSUE # 20 MAY 1988

well here we are again with another bumper issue of VZ link for your enjoyment.
It is good to see that we are getting a steady increase in membership.
We are also getting more younger members and that is really good for the hobby.
In this issue we have many fine items to help you use and understand your VZ.
Among them is a new item by Larry Taylor which will give you a better understanding of how the VZ and Epson printers handle Dot Matrix Graphics.

On page 9 are two Super COMPETITIONS which you may be interested in entering.

We also have two pages which are submitted by Eddie Tomes of Australia.
Eddie has compiled a list of all the COMMERCIAL Software which has been written for the VZ.
He has also made a couple of important statements which I indorse.
I have put these statements in Boxes to highlight them.
I have also included in this issue a program called spot-light written by Danny Todd.Danny is one of the younger members of the club that I mentioned above.

If you have just joined and are still waiting for your back issues please give me a ring and I will post them out.
(see back page for phone number)

The workshop/meeting will be held on 28th May starting at 10.15am.
You are all invited to attend and details are enclosed with this issue.
I will also be contacting some of you,to see if you can bring your equipment.If you wish to offer,to bring your VZ gear please ring me.

Some of you have been over for informal user sessions at my place.
You are most welcome,just ring first to make sure I'am home.
Would the person who offered me a copy of the Tennis game program please give me a call.
Any other local items for the newsletter would be great.

Last issue I sent out Subscription renewal reminders.
If yours are due don't forget to forward them to me.
To those that have,Thank you,these fund's help to cover the cost of the newsletter. Also the prizes for the competitions ...

So Until next issue or I see you at the Workshop/meeting..
Have fun with your VZ
Regards Peter

EDUCATIONAL PROGRAM BY P.LEON

The program below is designed to help learn ELEMENTS and their SYMBOLS. You can chose ELEMENTS, SYMBOLS or RANDOM of the two previous. Whichever you chose, you'll have to type in either the ELEMENT's name or it's SYMBOL. Even if you know the element's name the spelling has to be 100% correct. Have fun.

Only 20 elements have been included and you can add or replace with different elements. Do not forget to reDIMension if you add or subtract number of elements.

```
10 '*************************************************************
20 '*   TRAINING PROGRAM FOR LEARNING ELEMENTS AND SYMBOLS      *
30 '*   TOGETHER WITH PERCENT RIGHT SCORE --- BY PAUL LEON      *
40 '*************************************************************
50 :
60 POKE30744,96:DIM E$(20),S$(20):GOSUB100:GOSUB700:FORI=1TO20
70 N=RND(20):IFE$(N)<>""THEN70
80 READE$(N),S$(N):NEXTI:GOTO200
90 :
100 CLS:PRINT TAB(8)"ELEMENTS QUIZ"
110 PRINT" ==============================="
120 PRINT"  NUMBER OF QUESTIONS ANSWERED"
130 PRINT"  RIGHT >        WRONG >"
140 PRINT"  PERCENTAGE CORRECT -->"
150 PRINT" ==============================":RETURN
160 :
200 PRINT@226,"QUESTION #       WAS":PRINT@418,"QUESTION #"
210 FORI=1TO20:PRINT@429,USING" ##";I:GOSUB600
220 PRINT@482,"PRESS 'Q' TO QUIT";
230 IFEL$="E"THEN310ELSEIFEL$="S"THEN250
240 A=RND(2)-1:IF A THEN310
250 PRINT@290,"ELEMENT     ";E$(I)
260 PRINT@354,;:INPUT"SYMBOL    ";A$
270 IFA$="Q"THENGOSUB500:GOSUB600:GOTO370
280 IFA$=S$(I)THENGOSUB500ELSEIFA$<>S$(I)THENGOSUB510
300 NEXT:GOSUB600:GOTO370
310 PRINT@354,"SYMBOL     ";S$(I)
320 PRINT@290,;:INPUT"ELEMENT   ";A$
330 IFA$="Q"THENGOSUB500:GOSUB600:GOTO370
340 IFA$=E$(I)THENGOSUB500ELSEIFA$<>E$(I)THENGOSUB510
360 NEXT:GOSUB600
370 PRINT@440,"Y   ":PRINT@418,;:INPUT"LIKE TO TRY AGAIN ";T$
380 IF T$="Y"THENRUNELSEIFT$="Q"THENCLS:END:ELSE370
390 :
500 PRINT@247,"RIGHT":SOUND31,1:R=R+1:GOTO520
510 PRINT@247,"WRONG":SOUND15,4:W=W+1
520 PRINT@108,USING"##";R:PRINT@124,USING"##";W
530 PRINT@152,USING"###.##";R/I*100
540 PRINT@237,USING" ##";I:RETURN
590 :
600 FOR J=288TO388STEP32
610 PRINT@J,"                                ";:REM 32 SPACES
620 NEXT:RETURN
690 :
700 PRINT@290,"ELEMENTS - SYMBOLS - RANDOM":PRINT@379,"R"
710 PRINT@355,;:INPUT"SELECT  E OR S OR R   ";EL$
720 IFEL$<>"E"ANDEL$<>"S"ANDEL$<>"R"THEN700ELSERETURN
790 :
800 DATA HYDROGEN,H,HELIUM,HE,LITHIUM,LI,BERYLLIUM,BE
810 DATA BORON,B,CARBON,C,NITROGEN,N,OXYGEN,O,FLOURINE,F
820 DATA NEON,NE,SODIUM,NA,MAGNESIUM,MG,ALUMINIUM,AL,SILICON,SI
830 DATA PHOSPHOROUS,P,SULFUR,S,CHLORINE,CL,ARGON,AR
840 DATA POTASSIUM,K,CALCIUM,CA
```

STARTER HINTS FOR BEGINNERS.

So you have been smart enough to have purchased, borrowed or lucky enough to have been given a personal computer, a VZ colour computer at that. Perhaps you are having "teething" troubles and not been able to use the great little VZ with ease. It is a fact that a perfect computer operators manual for any computer is impossible to produce. Some user folk say that a particular manual is too simple and childish in its format. Others may say that it is too complicated and hard to follow. ***IT'S THESE FOLK THAT THIS ARTICLE IS AIMED AT TO HELP.***

The first thing that should be done is to affix the computer, tape cassette unit and power packs to a baseboard of some sort in a manner so that there is no movement of the pieces. 99% of problems are caused by loose pieces, in particular movement of the plug-in RAM memory expansion unit. The effects caused by loose pieces are computer "crashes", which means that the programme dissappears, the screen (VDU Video Display Unit) is full of graphic or garbage characters and text. Another is that programmes cannot be loaded into the VZ from tape, and so on. While the system is still in it's warranty period, use wire or sticky tape to hold the units firmly. When the warranty has expired you can then use a more permanent method such a screwed metal brackets.

TAPE LOADING.

Despite sales pitches to the contrary, most cassette recorders are NOT 100% satisfactory to use with most brand computers. This means that programmes can not be loaded the **first time,or perhaps not at all.The volume control has to be fiddled with and tried again and again until success may** be achieved. The old VIC 20 was designed to use only the Commodore cassette recorder (CTR), and although the VZ does have a specially designed CTR to be used, some folk opt to using an ordinary CTR with the resulting frustration and the early demise of the VZ to a cupboard. The special VZ DTR (Data Tape Recorder) shapes the computer signals before they are recorded and after they are are played back into the VZ, for 99.99% reliability.

If a "bad load" is affected at any part of the programme on the tape, the error message "LOADING ERROR" and "WAITING" below that is displayed in the left bottom of the VDU. You can rewind the tape to try again or if not, just press <CTRL><BREAK> and the "READY" message and flashing cursor will appear as normal.

DSE DEMONSTRATION TAPE.

I advise that you load the demonstration tape that comes with the VZ into the Data Cassette Recorder (DCR), rewind it to it's start and set the tape counter to 000 (zero). Load the programmes by pressing <CTRL><CLOAD><RETURN> and start the DCR and note on paper the DTR counter numbers where each programme starts. In all cases, if the LOADING message appears at number 045, as an example, on the DTR, then write that number down. If at another time you wish to load that programme again, wind the tape back to the start and set the counter to 000. Now wind the tape to about 044 not 045. This is because all programmes and data files have a "Header" on the tape. The "Header" gives various information to the VZ which is put on the tape in the recording mode. This "Header" lasts for about 3 to 4 seconds. So the tape must in a position for the VZ to receive the "Header", in this example about 044. Some folk instal a sound monitor to their VZ DTR so that this start "Header" can be heard so the tape can be started at the correct position.

To load a particular programme, it's correct file name must be typed in. Say you wish to load GAME TWO you type:- CRUN"GAME TWO"RETURN.

To print the above in instructions in the printed medium, it is denoted this way :- <CRUN>"GAME TWO"<RETURN>. That is if you type CRUN by the normal single key typing manner. If you type in the single key mode as written above each key it is :- <CTRL><CRUN>"GAME TWO"<RETURN>. The file name must be EXACTLY the same as on the tape "Header", and that means any full stops, spaces etc. If you don't know what the file name is then don't type any file name :- <CTRL><CRUN><RETURN> and the first programme that is received will be loaded, together with it's file name. If that is not the programme that you want then stop the DTR and press <CTRL><BREAK>. Wind the tape and try again at another position. If you find the programme, just let the tape run until it is loaded in. This is indicated by either the "READY" and flashing cursor, or the programme/game starting, depending on the type of programme.

There is no need to do a <NEW> to remove that portion of the previous programme that was just loaded before the tape was stopped. The VZ does that for you. Clever isn't it? We do have a Utility programme called FILESEARCH that prints on the VDU and/or printed on paper all the programmes on a tape or disc. It is up to the user to note the VZ DTR tape indicator numbers.

LISTING A PROGRAMME.

There are various ways that a programme can be saved. The letter before the colon and programme name (file name) can be any one of the alphabet. IE. LOADING T: GAME 2, or LOADING B: GAME 7. T and B are the most common. A programme with a T prefix may be listable. A programme with prefix B will not be listable. If you wish to LIST a programme, load it with the <CLOAD> command, not the <CRUN> command, which RUNs the programme straight away and you may not be able to <CTRL><BREAK> it.

To <LIST> it, press <CTRL><LIST><RETURN>. Remember, single key entries can be spelt out. In this case, L I S T. The <LISTing> will start at the lowest line number. If you wish to stop the <LISTing>, just press <SPACE>. To continue press <SPACE> again. The <LISTing> will continue to the last line number, which will be the highest number. If you wish to stop the <LISTing> completly to Edit it, then press <CTRL><BREAK>. Some programmes with a T prefix cannot be readily listed. Some of the earlier DSE. units are in this catagory so that when a <LISTing> is done only one line is printed:- XXX POKE30862,XXX:POKE 30863,XXX:USR(X)

There are other prefixes in use, F, K, S and D which is the normal one for a DATA file. W is used by the Editor/Assembler.

OTHER BRAND COMPUTER TAPES.

Another question often asked is if another computer's (ATARIE, COMMODORE, MICROBEE etc.) programme on tape can be loaded into the VZ. In short the answer is NO NO NO. BUT with a with special software Utility the anwer is YES. One case in point is that by using our VSOFTWAREZ TU18 LOAD XX-80 FILE unit, the early TRS 80 and SYSTEM 80 tape programmes can be loaded in. The programme can be <LISTed> and most probably will require you to change (EDIT) some of it. I do not know of any other such software for other computers, but would like to hear from anyone who does..

MEMORY SIZES.

If your computer has enough Memory, then it will run any normal programme written for the VZ200 or the VZ300. In other words, the VZ200 and VZ300 are entirely compatable, well ALMOST!!. There are a few isolated instances where the printer will not operate with the DSE. Editor/Assembler. Another is that the VZ300 may not operate correctly with the DSE. Word Processor.

But for all intents and purposes, the VZ200 and the VZ30 0 are compatable.

The confusion arrises where folk are not sure whether their computer's Memory is large enough. There is such a thing called a MEMORY MAP. This is printed in our ***VPROGRAMMEZ-VHINTZ-VHARDWAREZ Book,*** in the VZ Technical Manual or can be obtained by contacting Mr.Bob Kitch whose address is printed elsewhere in LE'VZ. It shows quite clearly what you need to know about memory size. Our software listings denote this by a code:- VZ1 VZ2 VZ3 VZ4 and VZ5.

The smallest memory size is the unexpanded VZ200 (VZ1), and the largest are both VZ's with a 64K RAM PACK (VZ5). Of course there are other sizes of memories, most being built up by folk who are technicaly capable. Such units can be supplied, or manufactured, please contact me.

Hopefully this will answer most of the "first time" questions asked by beginners to VZ computing. I may expand on the above in later LE'VZ magazines. If folk have any suggestions in this regard, please let me know.

J.D'Alton, LE'VZ.

(VZ) 04.02.88.

DOT MATRIX GRAPHICS by Larry Taylor

Whilst most EPSON printers and their innumerable emulators share a common set of print codes, there are often extensive variations to each printer command set. Competing brands, in an attempt to outdo the oppposition and distinguish themselves from other printers in their price range, will offer various enhancements over a base EPSON model. These may include a choice of typefaces and printstyles, letter quality print, or even increased speed. Some of these improvements are achieved through hardware modifications, such as increasing the number of pins in the printhead, whilst others are due to the software installed in the printer's ROM.

One area where software differences between printers can be immediately noticed, is in the way a single character is printed on paper. Each character printed, consists of columns of dots in series. The number of columns, and in particular the number of dots in each column, will vary according to the capabilities of the printhead in each printer. Software can, however, make a difference to the final appearance of the character. My first EPSON compatible printer was a BMC BX-80, which eventually gave way to an OLYMPIA NP165. A simple comparison of characters printed by these two machines showed there was a very obvious difference. The dot columns printed by the BMC are close together, whilst those displayed by the OLYMPIA are slightly spaced. This results in a character printed by the BMC being marginally narrower than that created by the OLYMPIA. This difference in dot density, is due to the control, which the ROM software exercises over the data, which ultimately operates the printhead. Most EPSON type printers possess two, bit image modes, that allow the user to directly control, which pins are fired in the printhead. All the characters produced by a printer are created using these bit image facilities. However, because the printer stores the machine code driver routines and the character shape tables (these are sets of numbers, which determine a character's shape) in its own ROM, it can churn out its own predesigned shapes considerably faster, than it can the individual bit image data transmitted to it by a host computer. The use of an inbuilt RAM buffer and a facility for downloading character data to the printer is an attempt to partly overcome the speed limitations of dot image processing.

The term, bit image, may need some explanation. A byte of data on a computer, such as the VZ, consists of eight bits. Each of these individual bits can only be equal to one or zero. To correspond to this, a printhead, in its simplest form, consists of 8 pins arranged vertically. When a single byte of data is sent to the printhead, the value of the individual bits will determine which pins are fired. For example, if bit 4 is equal to one then pin 4 will produce a dot on the paper. This is often referred to as the printer's graphics mode, because it can be used to produce shapes drawn by the user. The size and shape of the image to be produced is limited only by the width of paper the printer can handle.

The two, bit image modes on most EPSON type printers are usually referred to as single or normal density and double density. These terms, basically refer to the number of dots that can be printed across the page. In double density mode, twice the number of dots can be printed across the page as there would be in single density, however, since the paper width is the same, the dots have to be much closer together, giving the resulting image a much darker appearance. The actual number of dots printed varies from printer to printer. The EPSON standard usually specifies 480 dots in single density mode and 960 dots in double density. My BMC BX-80, however, does not conform to this and chooses, instead, to use 640 dots and 1280 dots respectively. The OLYMPIA, on the other hand, whilst conforming to the EPSON original, additionally offers, a further four 8 pin bit image modes and two, which utilise a ninth pin. One of the 8 pin modes is referred to as quadruple density and can print 1920 dots across the page. As well, this printer allows for 640 dot and 1280 dot modes to be accessed. These modes are cryptically referred to as CRT graphics, which I can only assume stands for Cathode Ray Tube, since the manual doesn't tell me. Not being familiar with the screen resolution of oscilloscopes, I can only guess that these dot widths are necessary to allow the full width of such images to be dumped to the printer. Although these modes haven't been assigned the standard EPSON codes, it does show that the OLYMPIA can emulate the BMC, if necessary.

This lack of conformity is what caused problems for me, when I came to write the screen dump routines for the VZ-EPSON Printer Patch. As we all know, the MODE(1) screen, on the VZ, is 128 pixels wide and there can only be four colours on the screen at any one time. To print a representation of the screen on paper, it is necessary to first work out a dot image to stand for each differently coloured pixel. Green was most easily represented as blank, whilst red was displayed as completely black. Yellow had to be shown as a slightly lighter image than that displayed for blue. I then had to decide on how many dots wide, I would make each pixel image. To calculate this, the number of pixels, 128, is multiplied by the number of dots in each printed image and then displayed as a total. The simple mathematics is displayed below.

128 X 3 dots = 384 dots

128 X 4 dots = 512 dots

I was unaware of the differences between printers outlined above, so in my earlier versions of the patch I opted for a 4 dot width, which meant that the screen dump was 512 dots wide. Since my BMC was capable of a 640 dot width and I was well within that, I experienced no problems. Not so, those purchasers of the patch who owned EPSON type printers, which conformed to the 480 dot standard. These people discovered, to their dismay, as I did after buying the OLYMPIA, that when they did a screen dump, 32 dots or 8 pixels from the right hand side of their picture didn't fit on the page. It was therefore necessary to make a small change to later versions of the patch (Version 1.4 being the most recent), so that the pixel images used were only 3 dots wide. This gave dumps with a total dot width of 384 dots. Solving this problem was easy, but because of the difference in dot density between the BMC and OLYMPIA, screen dumps done on the BMC now appear too narrow. I cannot see any simple solution to this problem, as it is controlled by the software within each printer's ROM. Nevertheless, I hope that this has helped to answer some of the queries I have had from people concerning operation of the screen dump.

In this next section, I will attempt to explain how the standard bit image modes of an EPSON compatible printer can

be accessed from BASIC. Because the method used for either mode is very similar, our discussion will focus on single density dot graphics. Various printer manuals display the necessary code in one or more of the following ways.

ASCII	ESC	K	+	n1 + n2
Decimal	(27)D	K(75)D	n1	n2
	$[27]_{10}$	$[75]_{10}$	[n1]	[n2]
Hexidecimal	(1B)H	(4B)H	(n1)	(n2)
	$[1B]_{16}$	$[4B]_{16}$	[n1]	[n2]

The first two values [27] and [75] put the printer into normal density bit image mode. The next two values, n1 and n2 tell the printer how many bytes of graphic data will be transmitted. They are arranged in low byte, high byte order. For example if we were doing a screen dump, each line would require 384 bytes of data. The low byte and high byte can be calculated in this way.

high byte n2 = INT(384 / 256)

low byte n1 = 384 - n2 * 256

Using this method n1 = 128 and n2 = 1. These values will, of course vary according to the number of bytes to be sent. What follows next, is a stream of graphics data, which, in the case of the example used above, will consist of 384 individual bytes. Provided the low byte, high byte values, when calculated out,

eg. no. of data bytes = n1 + 256 * n2

equals the number of graphics data bytes which follow, the design will be printed. If the printer does not respond by printing the graphic, it will usually mean that either the low byte - high byte values haven't been entered correctly, or that the printer has not yet received the expected number of data bytes and is waiting for them.

To illustrate the procedure, I will demonstrate how to print this heart shaped character ♥ using single density, bit image, dot graphics. This character consists of nine dot columns. Each column is represented by a single byte of data. In addition, every column is made up of eight rows, each of which is represented by a single bit within the byte of data. When reading a column from the bottom up we begin at bit 0, which has the value 1, and finish at bit 7, which is equal to 128. A single data byte is calculated by totalling these row values assigned to each of the dots printed in a column. Because the character consists of nine columns, it is represented by nine similarly calculated data bytes.

Bit	Value	
7 =	128	■■■■■■■■
6 =	64	■■■■■■■■
5 =	32	■■■■■■■■
4 =	16	■■■■■■■■
3 =	8	■■■■■■■■
2 =	4	■■■■■■■■
1 =	2	■■■■■■■■
0 =	1	■■■■■■■■

By thus totalling each column in turn, vertically, from left to right we arrive at the following byte values.

112, 248, 252, 126, 63, 126, 252, 248, 112

The number of data bytes to be sent to the printer to produce this particular character is nine. Represented in low byte - high byte form this is shown as,

low byte = 9

high byte = 0

So to produce the heart character, we send the following data to the printer.

Put printer in graphics mode :- 27,75,

Number data bytes (low byte - high byte) :- 9,0,

Data bytes :- 112,248,252,126,63,126,252,248,112

The easiest way to accomplish this from BASIC, would be as follows.

```
100 REM SEND EPSON CODE TO PUT PRINTER IN SINGLE DENSITY BIT
110 REM IMAGE MODE.
115 REM ***************************************************
120     LPRINT CHR$(27);"K";
125 REM ***************************************************
130 REM LOW BYTE - HIGH BYTE DATA TO TELL PRINTER HOW MANY
140 REM BYTES OF GRAPHICS DATA TO EXPECT.
145 REM ***************************************************
150     DATA 9,0
155 REM ***************************************************
160 REM NINE GRAPHICS DATA BYTES NECESSARY TO PRODUCE A
170 REM HEART SHAPED CHARACTER.
175 REM ***************************************************
180     DATA 112,248,252,126,63,126,252,248,112
185 REM ***************************************************
190 REM SOME VALUES SUCH AS 0,10,11,12,13 AND THE INVERSE
200 REM AND GRAPHICS CHARACTER VALUES FROM 128 TO 255 WON'T
210 REM REACH THE PRINTER IF WE LPRINT THEM, SO SEND THE LOW
220 REM BYTE - HIGH BYTE AND NINE DATA BYTES OUT THE PORTS.
225 REM ***************************************************
230     FOR T=1 TO 11:READ D: GOSUB 300:NEXT
240     END
245 REM ***************************************************
250 REM READ PORT ZERO TO CHECK IF PRINTER IS READY. WHEN
260 REM IT IS, SEND DATA OUT PORTS 13 AND 14.
265 REM ***************************************************
300     IF (INP(0)AND1)<>0 THEN 300 ELSE OUT 13,D:OUT 14,D
310     RETURN
320 REM ***************************************************
```

To further assist those of you, who may still be experiencing difficulty in getting your EPSON type printer to produce dot graphics, I have written a short DOT MATRIX GRAPHICS EDITOR in BASIC. This program enables a design to be drawn on the screen by moving a cursor within a grid, which is 8 rows high and 11 columns wide. Once the design is completed and the RETURN key pressed, the graphics data is automatically tabulated by the program. Following this, the printer is set to single density bit image mode and the eleven bytes of graphics data are sent to the printer, preceded, of course by the low byte - high byte equivalent of the number eleven.

Once the concept of producing images on the printer, using dot graphics, has been fully grasped and you have experienced the first joys of success, there remains only to experiment with your own designs. There are two ways, in which you can create your images prior to transferring them to the printer. The first is to do a drawing on squared paper, calculate the column values, create data statements and then transmit them to the printer. The second, possibly easier method, depending on the resolution required, is to use an EDITOR to draw on the HIRES screen. On completion, it can read the screen, compute the data and send it to your printer. Having used both approaches, I find an EDITOR easier. Don't despair if you don't experience immediate success. My early frustrations culminated in the successful creation of the VZ-EPSON Printer Patch.

```
100 REM##############################
110 REM# DOT MATRIX GRAPHICS EDITOR #
120 REM#       BY  LARRY TAYLOR     #
130 REM##############################
140 REM
150 REM#########################
160 REM#  EDITOR CONTROL KEYS  #
170 REM#########################
180 REM#  [M]    - CURSOR LEFT  #
190 REM#  [,]    - CURSOR RIGHT #
200 REM#  [.]    - CURSOR UP    #
210 REM#[SPACE] - CURSOR DOWN  #
220 REM#  [M]    - CURSOR LEFT  #
230 REM#  [L]    - INSERT BLOCK #
240 REM#  [;]    - RUBOUT BLOCK #
250 REM#[RETURN]- PRINT SHAPE  #
260 REM#  [C]    - CLEAR SCREEN #
270 REM#########################
280 REM
290 REM#########################
300 REM# INITIALISE VARIABLES  #
310 REM#########################
320    P=28672:D=60:E=207:DIMV(10)
330    POKE30744,0:CLS:COLOR2
340 REM#########################
350 REM# SET UP EDITOR DISPLAY #
360 REM#########################
370    PRINT@  0,"┌──────────────────────────────────┐";
380    PRINT@ 32,"│  DOT MATRIX GRAPHICS EDITOR   │";
390    PRINT@ 64,"└──────────────────────────────────┘";
400    FORT=3TO14:POKEP+T*32,154:POKEP+T*32+31,149:NEXT
410    PRINT@480,"└──────────────────────────────────┘";:POKEP+511,151
420    P=28844:PRINT@130,"BIT VALUE═══════════"
430    FOR T=7 TO OSTEP-1
440    PRINT@162+(7-T)*32,USING"##    ";T;:PRINTUSING"### ";2^T;
450    PRINT"│             │":NEXT
460    PRINT@427,"└─────────────┘"
470    M=X+Y*32:V=PEEK(P+M)
480 REM##################################
490 REM# READ KEYPRESS AND FLASH CURSOR #
500 REM##################################
510    A$=INKEY$:POKEP+M,V:FORT=1TOD:NEXT:POKEP+M,E
520    FORT=1TOD:NEXT:A$=INKEY$:IFA$=""THEN510
530    A=ASC(A$):IFA=67THENRUN
540    IFA=13THEN560
550    IFA<>32ANDA<>44ANDA<>46ANDA<>59ANDA<>76ANDA<>77 THEN510
560    POKEP+M,V:FORT=1TOD:NEXT
570    IFA$=CHR$(13)THENGOSUB690
580    IFA$="M":X=X-1:IFX<0 :X=0
590    IFA$=",":X=X+1:IFX>10:X=10
600    IFA$=".":Y=Y-1:IFY<0 :Y=0
610    IFA$=" ":Y=Y+1:IFY>7 :Y=7
620    M=X+Y*32
630    IFA$="L":POKEP+M,128
640    IFA$=";":POKEP+M,96
650    V=PEEK(P+M):GOTO510
```

CONTINUED NEXT PAGE

```
660 REM#########################
670 REM# CALCULATE DATA VALUES #
680 REM#########################
690     FORX=0TO10:FORY=0TO7:M=X+(7-Y)*32
700     IFPEEK(P+M)<>128THEN720
710     V(X)=V(X)+2^Y
720     NEXT:NEXT
730 REM#########################
740 REM#  PRINT GRAPHIC IMAGE  #
750 REM#########################
760     LPRINTCHR$(27);"K";
770     D=11:GOSUB790:D=0:GOSUB790
780     FORT=0TO10:D=V(T):V(T)=0:GOSUB790:NEXT:X=0:Y=0:RETURN
790     IF(INP(0)AND1)<>0THEN790ELSEOUT13,D:OUT14,D:RETURN
```

SEE PAGE 8 FOR ANOTHER EXAMPLE

TWISTER CORRECTION

In the March issue I gave you some changes that could be made to the New Zealand Twister program.
The Idea was to change it into a sliding block puzzle.

Unfortunately,I didn't read richard's letter correctly and as a result I may have put you wrong.
Richard has written again,so let's put things right.
(If your program crashed due to this error pse ring and I will give you a working copy of twister)

NOTES ON NZ TWISTER AGAIN

There are some differences in moving the squares in this new version of the game that Peter didn't mention and they are that with the NZ TWISTER you had the keys from 1 to 9 to twist the block of four squares with this one you only move one square from one place to another so therefore I have labled the 4X4 grid as below.....

A B C D

E F G H

I J K L

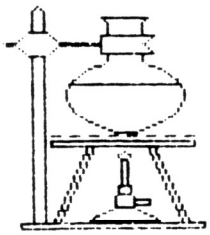
M N O P

So let us suppose after the mixing of the squares that F is blank you can then move either B,E,G or J into F, to do this there are two key strokes the first press the key you want to move, say we will move B into F so press B(beep) then F(beep), now B will be in F and B will be blank, you can <u>only</u> move into a blank square if you try to move into a square that is not blank it will not go, but you only have to enter the last key (the blank square) to correct the move.

Also peter said that ot change the puzzle picture just change the DATA statments, this is quite true but be careful as the first four DATA lines contain the m/c program that displays the data to the screen, to change this would make the program crash.

The four zero's(0) in line 406 is the start of the data for the blocks to make up the squares in the puzzle, there are sixteen blocks of sixteen bytes that make up the whole puzzle, and only these can be changed.

Richard Davis



```
5 DIMD(772)
8 LPRINTCHR$(27);"A";CHR$(7);
10 FORT=1TO772:READD:D(T)=D:NEXT
15 FORI=0TO11:LPRINT"   ";
18 FORX=1TO4:D=D(X):GOSUB50:NEXT
20 FORT=1TO64
25 D=D(4+I*64+T):GOSUB50:NEXT
35 LPRINT:NEXT:END
50 IF(INP(0)AND1)<>0THEN50ELSEOUT13,D:OUT14,D:RETURN
90 DATA27,75,64,0
200 DATA0,0,0,0,0,0,0,0,0,63,64,128,64,63,0,0,0,0,0,0,0,0,0,0,0
205 DATA0,0,0,0,0,0,8
210 DATA12,11,8,8,8,8,8,8,8,8,8,8,11,12,8,0,0,0,0,0,0,0,0,0,0,0
215 DATA0,0,0,0,0,0
220 DATA0,1,1,1,1,2,4,8,16,240,16,16,16,240,16,8,4,2,1,1,1,1,1
225 DATA1,1,1,1,1,1,1,1,15
230 DATA8,248,8,8,8,8,8,8,8,8,8,8,248,9,10,12,8,0,0,0,0,0,0,0,0
235 DATA0,0,0,0,0,0,0
240 DATA128,64,64,64,192,32,16,8,4,7,4,4,4,7,4,8,16,32,192,64,64
245 DATA64,64,64,64,64,64,64,64,64,64,248
250 DATA8,15,8,8,8,8,8,8,8,8,8,8,143,72,40,24,8,0,0,0,0,0,0,0,0
255 DATA0,0,0,0,0,0,0
260 DATA0,0,0,0,0,0,0,0,0,255,0,0,0,255,0,0,0,0,0,0,0,0,0,1,2,4
265 DATA8,16,16,32,32,32
270 DATA64,128,0,0,0,0,0,0,0,0,0,0,128,64,32,32,32,16,16,8,4,2,1
275 DATA0,0,0,0,0,0,0,0,0
280 DATA0,0,0,0,0,0,0,0,0,255,0,0,0,255,0,0,0,0,0,0,15,48,224,32
285 DATA32,32,32,32,32,32,32,32
290 DATA32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32
295 DATA32,32,32,32,224,48,15,0,0,0,0,0,0
300 DATA0,0,0,0,0,0,0,0,0,255,0,0,0,255,0,0,0,0,0,0,192,48,12,2
305 DATA1,0,0,0,0,0,0,0
310 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,2,12,48,192
315 DATA0,0,0,0,0,0
320 DATA0,0,0,0,0,0,0,0,0,255,0,0,0,255,0,0,0,0,0,0,0,0,0,0,0
325 DATA128,64,32,32,16,16,8
330 DATA8,4,4,2,2,2,2,2,2,2,2,4,4,8,8,16,16,32,32,64,128,0,0,0,0
335 DATA0,0,0,0,0,0,0
340 DATA0,0,0,0,0,0,0,0,0,255,0,0,0,255,0,184,168,168,168,40,40
345 DATA168,168,168,168,40,47,168,175,168,168,40
350 DATA40,168,168,168,168,41,42,169,168,168,168,40,40,168,175
355 DATA168,175,40,40,168,168,168,168,40,40,168,168,168,184,0,0
356 DATA0
360 DATA0,0,0,0,0,0,0,0,0,255,0,0,0,255,0,0,0,0,0,0,0
365 DATA0,0,3,28,227,28,224,0,0,0,0
370 DATA0,0,0,0,192,63,32,63,192,0,0,0,0,0,0,224,28,227,28,3,0,0
375 DATA0,0,0,0,0,0,0,0,0,0
380 DATA0,0,0,0,0,0,0,0,0,255,0,0,0,255,0,0,0,0,0,0,1
385 DATA14,113,142,112,128,0,0,0,0,0,0
390 DATA0,0,0,0,7,252,4,252,7,0,0,0,0,0,0,0,0,128,112,142,113,14
395 DATA1,0,0,0,0,0,0,0,0,0
400 DATA0,0,0,0,0,0,0,0,0,255,0,0,0,255,0,0,0,0,7,56,199,56,192
405 DATA0,0,0,0,0,0,0,0,0
410 DATA0,0,1,62,194,66,66,66,194,62,21,20,20,20,28,0,0,0,0,0
415 DATA192,56,199,56,7,0,0,0,0,0,0,0
420 DATA0,0,0,0,0,7,5,5,5,253,5,5,5,253,5,5,29,229,29,229,5,5,5
425 DATA5,5,5,5,13,29,45,45,77
430 DATA77,141,13,13,13,13,13,13,13,13,13,141,77,77,45,45,29,13
435 DATA5,5,5,5,5,229,29,229,29,5,5,5,5,7
```

COMPETITION'S

COMPETITION 1

Write a program for the newsletter.It can be as nutty as you like. The program must work and must be original.It does'nt have to be long. Easy you say!!.There is a catch.
The program must be set out correctly and must have a statement or rem attached to each line that explains what the line does.

The prize for the best item submitted will be:
A box of disk's
Or 5x10min tapes.
Plus a suitable momento to go on your mantle.

All item's must be in before May 28th.
Judgeing will take place at the meeting/workshop on that date.

COMPETITION 2

The prize for this Competition is a Desk top Disk Organizer/card file. Valued at upto $60.00

All you have to do to win this one is:
Write a program that is suitable for sale,to raise fund's for the club.

The program can be either a Game or utility.It would have to be fairly good and must be suitable for tape and disk.

Judgeing will also take place on May 28th.

If you can't make it to the workshop you can still enter.
If you are lucky enough to win either prize,it will be posted out.

Z80 ASSEMBLY

After looking at addition with eight-bit numbers last month. We'll take a look at subtraction and the other mathematical operations this month.

Subtraction with eight-bit numbers is relatively easy. The Z80 supports a couple of instructions for this purpose, Namely SUB and SBC.
SUB has the form SUB s ,where 's' is a specified operand such as another of the eight-bit registers,a raw number,or even an indexed address.

In all cases,this operand is subtracted from the contents of the accumulator (A register),the result being placed in it after subtraction.

CONTINUED NEXT PAGE

For example....SUB 10H would subtract 10H from the accumulator.
SUB B would subtract the contents of B from A.
SUB (HL) would subtract the contents of the address pointed to by HL from A.

Now that we understand the importance of the Status Flag,it's helpful if we are told what effect any new instruction will have on it,if any.
After a SUB is executed,the zero flag is set according to whether the result was zero or not.The carry flag is set if there is a carry into the 'ninth bit'of the accumulator(i.e.the carry flag).
This would generally indicate if a negative results from the subtraction.The N flag is set to '0'.
The other subtraction instruction provided for eight bit operations is SBC.
This is virtually the same as the SUB instruction,except that the carry flag is first added to the operand to be subtracted.
The N flag is set to '1', and the other flags are set as they were for SUB.

The best way to determine whether the result of a subtraction is negative is to use the sign flag,which is effected in both of the above instructions.
Here's a routine to determine whether 'A' contains a number less than 'B'::

```
SUB B       ;subtract 'B' from 'A'
JP M,EXIT   ;exit if 'A' is a negative ('minus') number
```

You are probably wondering what use the SBC and ADC instructions have - why bother to add the carry on? These instructions are used in what is called 'multiple precision' operations. These involve numbers made up of several bytes, and we'll be looking at them later.

Just as the Accumulator was used as the basis for the eight-bit operations, the HL register pair is used in 16-bit addition and subtraction. For addition, ADD HL,ss and ADC HL,ss,are provided. In both cases, 'ss' must be another sixteen-bit register pair (i.e. BC. DE, HL or SP). The normal rules for adding the carry apply. Another add is offered for the IX and IY registers - ADD IX,ss and ADD IY,ss. For subtraction, only one instruction is offered - SBC HL,ss. As you can see, the HL register is the main register pair used in 16-bit arithmetic. At this stage, we're not going to go into handling numbers greater than 65535, as this requires a lot of work for what is rarely used by the hobbiest assembly language programmer. What is of use, however, is a basic understanding of the principles involved in multiplication and division.

The first method of multiplication with assembly is the Successive Addition system. Remember that we are limited to adding and subtracting at the processor level. In this system, the number to be multiplied (the multiplicand) is added to itself repeatedly, the number of times corresponding to the multiplier. For example, suppose we want to find '3 times 10' - this could be done in the following manner :

```
LD DE,10      ;set multiplicand
LD HL,0       ;set HL=0
ADD HL,DE     ;HL=DE * 1
ADD HL,DE     ;HL=DE * 2
ADD HL,DE     ;HL=DE * 3
```

The number (10) is effectively multiplied by three by repetative adding.

A more versatile example using the DJNZ instruction would be:

```
        LD DE,10
        LD HL,0
        LD B,3        ;Set multiplier
LOOP    ADD HL,DE
        DJNZ LOOP     ;And repeat 'B'times
```

In this case,the multiplier is more easily set by loading it into B in the third line.
Remember that in both cases,the result of the multiplication is left in HL.

The next method available to carry out multiplication is called the 'Shift and Add Multiply'.This is really just an extension of the successive add approach.
For example,suppose we want to multiply a number by 10.
Let the number=N. Now 10*N is equal to 8*n+2*n.
Multiplying by powers of two is relatively easy:Here's how we multiply the IX register by 10 :

```
        ADD IX,IX     ;double IX
        PUSH IX       ;save 2*IX on stack
        ADD IX,IX     ;now double again (*4)
        ADD IX,IX     ;now IX has been *8
        POP DE        ;DE=IX*2
        ADD IX,DE     ;IX=IX*8+IX*2=IX*10
```

Note that these methods of multiplication become fairly slow with large multipliers. However, they should provide an introduction to multiplication with Assembly. Once again, more complex methods go beyond the scope of this article, but if you are interested, consult a good Z80 software manual. No doubt you'll need more complex routines some time, so we're going to 'cheat', and look at routines already written for us in the Microsoft Basic ROM. But first, let's look at some simple division.

The simplest approach to division is by the 'Successive Subtract Method'. It divides a 16-bit dividend (the number to be divided) in HL, by a 16-bit divisor (number to be divided by) in BC. The process is repetitive - each time the divisor (BC) is subtracted from a dividend (HL), with DE counting the number of 'subtractions'. As soon as the result is negative, the routine exits, with DE equal to a whole number answer (the quotient). Note that the remainder could be found by adding the residue in HL to BC. Note also that the count (DE) starts at -1, as the first INC DE will always occur.

```
      LD HL,DIVD   ;HL=dividend
      LD BC,DIVS   ;BC=divisor
      LD DE,-1     ;start of count
LOOP  OR A         ;zero Carry flag
      SBC HL,BC    ;successive subtract
      INC DE       ;bump count (quotient)
      JP NC,LOOP   ;repeat if not negative (no carry)
                   ;else DE=quotient
```

Don't forget to replace DIVD & DIVS with the correct values. Once again, many more division routines are possible, but we wont go into them here. Next month we look at the ROM arithmetic routines, and at the different ways in which numbers are stored by them. Looking further into the future, once the theoretical concepts are out of the way, we'll begin taking practical examples of Assembly Language routines for the VZ specifically.

HI RES CHARACTER'S

WRITTEN BY RICHARD DAVIS

If you would like to use the HI RES letters and figures as used in my WORLD TIMES program,in a program of your own then here is the Subroutine to enable you to do so.
I will also explain how you can add all the rest of the ASCII characters if you wish,or any other HI RES character you may like.

The first part of the program from line 5-30 is a short routine to input some numbers or characters and then a Gosub calls the subroutine at 200 to display them in HI RES.
Being in basic it is very slow but at the same time can be effective for some displays.
If you want to try some other characters of your own then we will have to look at how the letters are made up.
Let's take the letter "A" as a example.
If you look at line 910,you will find:

```
910 E$="24144444114114444"
```

This is the data for the letter "A".
The letter is in a three by five grid as indicated by the first two numbers 2 and 4 (you start to count from zero).
Then the next fifteen numbers(the one's and four's) are the colours that you want the pixels to be.

If you add more ASCII characters to the list you will have to change lines 222 and also 235 as follows:
In line 222 Q=Q-55,the ASCII code for"A"is 65 so 65-55 gives us 10.
E$(10) gives us the data for the letter "A",so if you insert more ASCII characters in front of the alphabet this part of the line will have to be altered to suit the new position in the array.

Also line 235 looks for "M" and "W" to move the cursor along an extra two places,as these two letters are in a 5x5 grid.
We will have a look at line 910 again,but this time broken up into sections to get a better idea of how it goes together:

24 14444 41141 14444

	0	1	2
0	1	4	1
1	4	1	4
2	4	1	4
3	4	4	4
4	4	1	4

If you follow the 4's in the grid they make the letter "A"

If the first two numbers were 99 then you would have a ten by ten grid and so on.
But as this would mean 100 bits of data to form your characters,you would have to put the first half in one variable and the second half in another.
You would then need a line such as E$=Y$+Z$.
But just have a go at making up the rest of the ASCII set first and see how you go at that.

HI RES CHARACTERS CONTIN:-

```
1 CLEAR 1000:DIME$(36):GOSUB 900:CLS
4 PRINT@103,"  HI RES CHARACTERS":PRINT@135,"     BY PAUL DAVIS "
5 FORW=1TO2000:NEXT
15 CLS:PRINT@33,"NOTE <SPACE> TO ABORT MODE(1)"
17 FORP=1TO3:PRINT:NEXT
19 INPUT"WRITE SOMETHING";E$
25 MODE(1):X1=5:Y1=20:GOSUB200
30 I$=INKEY$:K$=INKEY$:IFK$=CHR$(32),GOTO15ELSE30
200 REM LETTERS
212 L=LEN(E$)
220 FORU=1TOL
221 L$=MID$(E$,U,1)
222 Q=ASC(L$):Q=Q-55
223 IF Q<0THEN X1=X1+4:GOTO240
230 N=0:A$(0)=E$(Q):GOSUB1010:N=1:Y1=20:GOSUB1100:X1=X1+4
235 IFQ=22 OR Q=32 THENX1=X1+2
240 NEXT
250 RETURN
900 E$(0)="24444444111444444"
901 E$(1)="24111114444411111"
902 E$(2)="24414444141444414"
903 E$(3)="24414144141444444"
904 E$(4)="24444411114111444"
905 E$(5)="24444144141441444"
906 E$(6)="24444444141441444"
907 E$(7)="24411114111144444"
908 E$(8)="24444444141444444"
909 E$(9)="24444144141444444"
910 E$(10)="24144444114114444
911 E$(11)="24444444141414141
912 E$(12)="24444444111441114
913 E$(13)="24444444111414441
914 E$(14)="24444444141441414
915 E$(15)="24444444141141411
916 E$(16)="24444444111441444
917 E$(17)="24444441141144444
918 E$(18)="24411144444441114
919 E$(19)="24111441111444444
920 E$(20)="24444441141144144
921 E$(21)="24444441111411114
922 E$(22)="44444444111144444441111144444
923 E$(23)="24444444111144444
924 E$(24)="24444444111444444
925 E$(25)="24444444141144411
926 E$(26)="24444414114444441
927 E$(27)="24444444141114144
928 E$(28)="24141144141441141
929 E$(29)="24411114444441111
930 E$(30)="24444441111444444
931 E$(31)="24444411111444441
932 E$(32)="44444441111444444411114444 44
933 E$(33)="24441441141144144
934 E$(34)="24444111144444411
935 E$(35)="24411444141444114
960 RETURN
1010 X=VAL(LEFT$(A$(0),1)):Y=VAL(MID$(A$(0),2,1))
1020 FOR I=0 TON:A$(I)=RIGHT$(A$(I),LEN(A$(I))-2)
1030 FOR J=0 TOX:FOR K=0 TO Y:T%(I,J,K)=VAL(LEFT$(A$(I),1))
1040 IF LEN(A$(I))=1 THEN 1060
1050 B$=RIGHT$(A$(I),LEN(A$(I))-1):A$(I)=B$
1060 NEXT:NEXT:NEXT:RETURN
1100 FOR I=0 TO Y:FOR J=0 TO X:COLOR T%(N-1,J,I)
1110 SET(X1+J,Y1+I):NEXT:NEXT:RETURN
```

' COMMERCIAL ** SOFTWARE '
For The
Dick Smith Electronics
VZ 200/300 Colour Computer

Program Name	Program Distributor
EDUCATIONAL PROGRAMS	
Elementry Geometry	DSE
Flashcard. 1 & 2	DSE
Flashcard 3 & 4	DSE
Maths Armada	DSE
Metric Spycatcher	DSE
Music Writer	DSE
Speed Reading	DSE
Statistics I	DSE
Statistics II	DSE
Spellomatic 1 & 2	DSE
Spellomatic 3 & 4	DSE
Typing Teacher	DSE
Whizkid Spycatch	DSE
Word Matching	DSE
Speed & Accuracy	LYS
Study Aid	LYS
Block Puzzler	VSW
Camping	VSW
Coordinates	VSW
Edudisk	VSW
European Capitals	VSW
Keyboard	VSW
Maths Countdown	VSW
Maths	VSW
Meatpies	VSW
Microscope	VSW
Name the Towns	VSW
Plus & Minus	VSW
Queensland	VSW
Tower Of Hanoi	VSW
Wordmatching	VSW
Radioman Pack	HIR
Parent/Teacher Pack	HIR
Morse Code	MIL
Code Maker/Break	McC
Maths Pack	McC
Number Challenge	McC
Word Challenge	McC
Block Puzzler	LAS
Coordinate	LAS
Edudisk	LAS
Keyboard	LAS
Maths Countdown	LAS
Meatpies	LAS
Microscope	LAS
Name The Towns	LAS
Snertle	LAS
Tower Of Hanoi	LAS
Type Teacher	BNE
GAMES PROGRAMS	
Snake Maze	LOW
Air Traffic Controller	DSE
Asteroids	DSE
Attack Killer Tomato	DSE
Backgammon	DSE
Biorythm	DSE
Blackjack	DSE
Checkers	DSE
Chess	DSE
Circus	DSE
Crash	DSE
Dawn Patrol	DSE
Defence Penetrator	DSE
Digout	DSE
Dracula's Castle	DSE
Duel	DSE
Dynasty Derby	DSE
Formula One	DSE
Galaxon	DSE
Ghost Hunter	DSE
Hamburger Sam	DSE
Hangman	DSE
Hoppy	DSE
Horse/Rabbit Racing	DSE
Invaders	DSE
Kamakazi	DSE
Knights & Dragons	DSE
Ladder Challenge	DSE
Learjet	DSE
Lunar Lander (Laser)	DSE
Lunar Lander (VZ300)	DSE
Matchbox	DSE
Maze Of Argon	DSE
Missile Attack	DSE
Othello	DSE
Penguin	DSE
Pharoah's Curse	DSE
Planet Patrol	DSE
Poker	DSE
Slot Mach/Russian Rou	DSE
Space Ram	DSE
Star Blaster	DSE
Super Snake	DSE
Tennis	DSE
Tennis / Golf Lesson	DSE
VZ-Panic	DSE
Adventure Island	LYS
Alien Escape	LYS
Alien Rain	LYS
Asteroid Field	LYS
Beat The Bug	LYS
Castle	LYS
Caverns Of Venus	LYS
City Blitz	LYS
Convoy	LYS
Cub Scout	LYS
Enchanted Princess	LYS
Goldmine Adventure	LYS
Graphic Golf	LYS
Haunted House	LYS
Land Of Scrod	LYS
Little Packer	LYS
Lost Sword	LYS
Melbourne Cup	LYS
Mousetrap	LYS
Pharoah's Tomb	LYS
Pyramid Adventure	LYS
Russian Roulette	LYS
Slot Machine	LYS
Ten Pin Bowling	LYS
Triffids : 2040 AD	LYS
Two Up	LYS
Ufo's	LYS
VZ Star Trek	LYS
VZ Star Trek II	LYS
Yahtzee	LYS
Boskone Alert	COS
Defence Penetrator	COS
Morgoth	COS
Rally Racer	COS
Road Warrior	COS
Air Traffic Controller	VSW
Blackjack	VSW
Escape River	VSW
Flight Deck	VSW
Flying Saucer	VSW
Haunted Mansion	VSW
Learjet	VSW
Mansion / Nova	VSW
Monopoly VZ	VSW
Poker Machine	VSW
Scotland Yard	VSW
Trivial Cult	VSW
Word Squares	VSW
Biorythm	MIL
Pak Man	MIL
Super Invaders	MIL
Airlocks Maze	CEL
Battle Galaxy	CEL
Battle Of Vietnam	CEL
Cataclysm Catapillar	CEL
Conan The Barbarian	CEL
Galactic Fighter	CEL
Ghoulbusters	CEL
Jailhouse Breakout	CEL
Lunar Landers 2020	CEL
Minatour Periless	CEL
Reactor Ddestruction	CEL
Royal Poker	CEL
Thermonuclear War	CEL
Titanium Triology	CEL
3D Merciless Maze	CEL
Golf	CAR
Skiing	CAR
Star Pilot	CAR
Target	CAR
Astro Save	TAI
Bright Idea	TAI
Cavern Chaos	TAI
Circuitry	TAI
Cross Over	TAI
Enervac	TAI
Flag Climber	TAI
Fighter	TAI
Fuse	TAI
Ghostly Maze	TAI
Grave Digger	TAI
Hunter	TAI
Missile Safe	TAI
Mower Mania	TAI
Pattern Guess	TAI
Phantom Driver	TAI
Piano Songs	TAI
Save Me	TAI
Shark Bait	TAI
Shoot	TAI
Snake	TAI
Snake Trails	TAI
Snake Trap	TAI
Three Hundred	TAI
Word Teaser	TAI
Enchanted Princess	McC
Hi - Jinks	McC
Gamblers Pack	McC
Graphic Golf	McC
Sultans Palace	McC
Ten Pin Bowling	McC
VZ Racer	McC
Star Treker	APE
The Lost Crown	APE
Blackjack Royale	VDU
Castle Greystone	VDU
Galctic Empires	VDU
Haunted Mansion	VDU
Knights Quest	VDU
Scotland Yard	VDU
Snake Duel	VDU
Stellar Wars	VDU
Trivial Cult	VDU
VZ Monopoly	VDU
Air Traffic Controller	LAS
Bust Out	LAS
Cosmic Rescue	LAS
Escape River	LAS
Keno	LAS
Learjet	LAS
Light Pen Puzzle	LAS
Magnum Quest	LAS
Solo Battleships	LAS
3D Maze	LAS
VZ Werm	LAS
Teacher Zapper	SOR
Dracula's Maze	BNE
Octapus	BNE
Base Defender	LYO
Memory	LYO
To The Hills	LYO
Graphics 1	LYO
Graphics 2	LYO
Krichon	LYO
Rally	LYO
Draw	LYO
Venture	LYO
Air Traffic Controller	BJE
Asteroids	BJE
Blackjack	BJE
Checkers	BJE
Circus	BJE
Defence Penetrator	BJE
Galazian Adventure	BJE
Ghost Hunter	BJE
Golf Lesson	BJE
Hangman	BJE
Hoppy	BJE
Invaders	BJE
Knights & Dragons	BJE
Ladder Challenge	BJE
Learjet	BJE
Panic	BJE
Planet Patrol	BJE
Poker	BJE
Rally X	BJE
Star Blaster	BJE
Super Snake	BJE
Tennis	BJE
BUSINESS PROGRAMS	
Database	DSE
Discounted Cashflow	DSE
Financial Cashflow	DSE
Mailing List	DSE
Matrix	DSE
Portfolio Management	DSE
Word Processor	DSE
Card File	LYS
Check Record	LYS
Decision Evalulator	LYS
Lemonade Stand	LYS
Master Finance	LYS
Metric Conversions	LYS
Racing Selector	LYS
Cash Book Ledger	VSW
Cheque Ledger	VSW
Database VZ	VSW
Le'VZ D'base	VSW
Le'VZ Statement	VSW
Mailing List	VSW
Quickwrite	VSW
Laser Calc	LAS
Mark 64 Database	LAS

Address Book	DAW
Data Master	DAW
Word Processor	DAW
Electronic Aid	APE
Finance Helper	APE
Matrix Utilities	APE
Super Word Processor	BRO
VZ Database	VDU
P.I.N. NBR. File	BNE

****

UTILITY PROGRAMS

****

Disassembler	DSE
Editor Assembler	DSE
Hex Utilities	DSE
Intro. To Basic	DSE
Sprite Generator	DSE
VZ-Mon 80	LYS
Hackers Delight	CEL
Program Copier	CEL
Screen Drawer	CEL
Array / Restore	VSW
Cmerge/Delete/Renum	VSW
Colour Graphics	VSW
Copy / Protect	VSW
Disassembler	VSW
Disc Guard	VSW
Disk Copy	VSW
Disk Ops 1	VSW
Disk Ops 2	VSW
Editor Assembler	VSW
Extended Basic (W.O.)	VSW
File Search	VSW
Load XX80 Files	VSW
Monitor / Debugger	VSW
Protect	VSW
Search Tape	VSW
Utilitys	VSW
VZ Epson Printer Patch	VSW
VZ Extended Basic(S.O.)	VSW
Weaving Drafts	VSW
VZ Debug	MIL
Graphics Pack	MIL
Art Gallery	LAS
Colour Hardcopy	LAS
Copy/protect	LAS
Diskguard	LAS
Disk Tool Kit	LAS
Editor/Assembler disk	LAS
Epson Printer Patch	LAS
Extended Basic (G.W.)	LAS
Fast Disk & Menu	LAS
Filesearch	LAS
Magic Paint	LAS
VZ Basic Compiler	LAS
VZ Convert	LAS
Colour Graphics	PAM
Monitor / Debugger	WOB
Renumber / Merge	WOB
VZ Extra Commands	LEH
Disk Menu	BNE
Poster Shop	BNE
Art Gallery	BNE
Tape I.D.	BNE
Tape / Disk List	BNE
Program Copier	SAR
Retrieval Program	BRO
Copier	SUN
VZ Chip 8	GRI
Graf Star	VDU
Name Star	VDU
Shop Star	VDU

****

SOFTWARE DISTRIBUTORS DECODING

****

APE	Ape Software
BJE	S.Bjelic
BNE	Brisbane VZ Workshop
BRO	G.Browell
CAR	Adam Carter
CEL	Celestron
COS	Cosmic
DAW	L.Dawson
DSE	Dick Smith Electronic
GRI	Griffin
HIR	J.Hirst
LAS	Laserlink
LEH	G.Lehman
LOW	B.Low
LYO	R.Lyons
LYS	Lysco
McC	G.McCleary
MIL	R.S.Miller
PAM	Pam Software
RHO	C.Rhodes
SAR	N.Sarafoudis
SOR	J.Sortland
SUN	E.Sundstrup
TAI	L.Tait
VDU	VZ Down Under
VSW	Vsoftwarez
WOB	W.Obrist

****

A number of programs appear under several suppliers and are sold on behalf of the one writer over a period of years. Also several different versions may have been available as the programs are improved as time goes by.

****

Some Software companies have either closed or are now distributing under new names.

****

LYSCO ....Stopped selling VZ software 31JUL 86 but no further word on new distributor for these programs.

CELESTRON ....Also N. Sarafoudis . These programs very hard to obtain even with a personal approach .

COSMIC .....Appears to have been taken over by DSE.

R.S.MILLER ....Closed down .

J.HIRST .... Also traded as HI-COM software.

BJELIC ... Most of this software appears to be pirate copies.

****

** CORRECTION .....**

VSW program listed as FLIGHTDECK appears to be incorrect name for Air Traffic Controller.

****

PLEASE NOTE ################

As Compiler of this list I do not hold copies of , Or have distribution rights for all of these programs but will assist in contacting the applicable Software House if required. The addresses are not included in this list as a number have changed or are no longer trading and I wish to prevent as many problems as possible for users in obtaining the programs of their choice by giving the latest information known.

****

To avoid all the mentioned software being lost to the future VZ user it would be an advantage to have a Central Llibrary to hold a copy of these programs so that they would be available when present support is no longer available. This would be done only once copyright restrictions have passed or writers permission has been obtained . Appropiate charges would then be applied to cover expenses of tapes or disks and postage etc.

Please advise me of your thoughts on this .

****

I would be interested to hear of any new or old software that is not listed so this catalogue can be updated at regular intervals . Please advise program name and also name and address of writer / distributor to enable correct records to be kept.

****

Compiled origionally :-
July 1986.
Updated :-
December 1986.
June 1987.
November 1987.

****

By:-.
Eddie Tomes
3 Kilkenny St.
Capalaba
Qld. 4157
Telephone 07 - 3902797 H

****

SPOT LIGHT

WRITTEN BY DANNY TODD

```
5 CLS:GOSUB600
10 CLEAR200:CLS
20 DIMA$(7):A=1:G=0:S=0
25 FORI=1TO7:READA$(I):NEXTI
30 X=0:Y=12:B$=A$(A)
40 F=0:N=0:NN=0
45 L=0:C=0:TC=18:C1=0
50 GOSUB380:GOSUB340
80 K$=INKEY$:I$=INKEY$
90 IFI$="M"THENNN=N-1
100 IFI$=","THENNN=N+1
110 IFNN>19THENNN=19
120 IFNN<0 THENNN=0
130 IFNN=10ANDF=0THENF=1
140 IFNN=0 ANDF=1THENF=2
150 GOSUB340
160 IFN<>NNTHENS=S+1
170 N=NN:G=G+1
180 GOSUB400
190 IFMID$(A$(A),N+1,1)=" "ANDL=1THENGOTO240
200 FORT=0TO5:NEXTT
210 IFF<2THENGOTO80
220 A=A+1
230 GOTO30
240 X=4:Y=1:B$="  YOU HAVE BEEN SEEN"
241 SOUND5,6;10,6;17,4;26,2:CLS
250 GOSUB380:PRINT"    "
260 PRINT"       YOU SCORED ";INT((A*1+S/G)*1)
270 PRINT:PRINT"       [illegible] [illegible] (Y/N)"
275 S$=INKEY$:C$=INKEY$
280 IFC$="Y"THENGOTO10
285 IFC$="N"THEN290ELSE275
290 CLS:PRINT"      BYE.....":FORW=0TO1000:NEXT:CLS
295 END
300 Y=3:X=10:B$="[illegible]":REM SHIFT J
310 GOSUB380:RETURN
320 X=10:Y=3:B$="  "
330 GOSUB380:RETURN
340 X=N:Y=13:B$="  "
350 GOSUB380
360 X=NN:B$="U"
370 GOSUB380:RETURN
380 PRINT@Y*32+X,B$;
390 RETURN
400 IFL=1THENC=C+1
410 IFC=TC THENL=0:TC=INT(RND(1)*5+(5-A)):GOSUB320
420 IFL=0THENC1=C1+1
430 IFC1=TCTHENL=1:C1=0:C=INT(RND(0)*5+(5-A)):GOSUB300
440 RETURN
480 DATA"[illegible]"
490 DATA"[illegible]"
500 DATA"[illegible]"
510 DATA"[illegible]":REM ALL SHIFT + Z
520 DATA"[illegible]"
530 DATA"[illegible]"
540 DATA"[illegible]"
```

```
600 PRINT@73,"[redacted]"
610 PRINT@104,"(BY DANNY TODD)"
612 PRINT"   IN THIS GAME YOU MUST TEST"
615 PRINT"   YOUR REACTIONS OR REFLEX'S"
620 PRINT"   AVOID THE SPOTLIGHTS BEAM":PRINT
625 PRINT"   USE <ARROW> KEYS TO MOVE"
627 PRINT"         LEFT AND RIGHT     ":PRINT
630 PRINT"   TRY NOT TO BE IN THE OPEN"
635 PRINT"   WHEN THE SPOT LIGHT IS ON"
640 PRINT"   YOU MUST GET THERE "
645 PRINT"            AND BACK TO SCORE":PRINT
650 PRINT@487," [redacted]";
655 C$=INKEY$:S$=INKEY$
666 IFS$=" "CLS:RETURN
670 GOTO655
```

VZ GAMERS HIGH SCORE LIST

This list is for New zealand VZ Game players.

Any VZ game is eligible to be entered on this list.
To register your high score all you have to do is send it in with the following details.

:Your Name
:Game Name
:Score and Date

Sign it,also get a witness to sign to verify your score.
This list rely's on your Honesty.
All scores submitted will be compared with Scott's list in VZ Down Under Newsletter. (see april issue)
If you do get a higher score then that shown on his list, I will forward details to scott .

GAME	SCORE	NAME
MISSILE ATTACK	37490	PETER HILL
GALAXON	12510	PETER HILL
GHOST HUNTER	12690	PETER HILL

P.O.BOX 1972 C.P.O AUCKLAND N.Z

(C) AUCKLAND VZ 300/200 COMPUTER USER'S CLUB 1986

ALL MATERIAL IN THIS NEWSLETTER IS SUBJECT TO COPYRIGHT..
CONTRIBUTED PROGRAMS & ARTICLES CAN BE REPRODUCED ON THE UNDERSTANDING THAT THEY ARE FOR THE PRIVATE USE OF SUBSCRIBERS ONLY.
COPYRIGHT IS ALWAY'S RETAINED BY THE AUTHOR.
THIS CLUB WILL NOT BE HELD RESPONSIBLE FOR THE BREECHING OF COPYRIGHT CAUSED BY THE PUBLICATION OF ANY MATERIAL SENT IN UNDER A FALSE NAME.

YOU ARE WELCOME TO PHONE ME BETWEEN 6.30 PM AND 9 PM ----PHONE 673-967